# Scaling Techniques for Large Markov Decision Process Planning Problems

Terran Lane & Leslie Pack Kaelbling

Artificial Intelligence Laboratory
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139

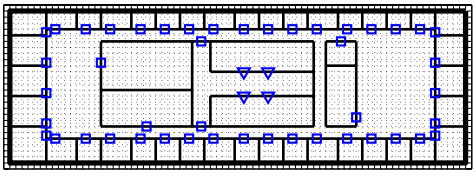
http://www.ai.mit.edu

**Planning in Large Domains:** The Markov decision process (MDP) formalism has emerged as a powerful representation for control and planning domains that are subject to stochastic effects. In particular, MDPs model situations in which an agent can exactly observe all relevant aspects of the world's state but in which the effects of the agent's actions are nondeterministic. Though the theory of MDPs is well developed and exact planning algorithms are known [3], these methods do not scale to the exponentially large state spaces that are commonly of interest in AI problems. In this project, we are examining approaches to reducing the complexity of MDP planning techniques in such large state spaces with an emphasis on classes of problems that arise in mobile robotics applications.

**Markov Decision Processes for Robotic Planning:** A Markov decision process describes a synchronous control process with four components: a *state space* that specifies all possible configurations of the system (e.g., the position of a robot in a map), an *action space* that lists the primitive actions available to the controller (e.g., a robot's movement and manipulation commands), a *transition function* that specifies the, possibly stochastic, outcomes of taking each action in any state, and a *reward function* that defines the goals of an agent in this space. In practice, the state space is often expressed as a cross-product of *state variables* (e.g., $x$ and $y$ coordinates or a list of what the agent is holding), yielding a space that is exponentially large in the number of such variables. It is assumed that the state space is both *complete*, in that all relevant variables are encoded, and *fully observable*, in that the agent can directly sense all state variables.

The goal of planning in the MDP framework is, given a complete description of an MDP, to develop an optimal *policy* that specifies which action to take in any state. An optimal policy is one that maximizes expected accumulated reward over the lifetime of the agent. While it is often possible to compactly describe even large MDPs with dynamic Bayes network representations of the transition function [1], a full description of the policy may still require a complete enumeration of the state space.

In mobile robot applications, we can formulate MDPs that represent the dynamics of the robot's environment as well as the tasks that the robot is to accomplish. Part of an example target problem is displayed in Figure 1, which shows a low-resolution discrete map of one of the ten floors of the building housing the MIT AI Lab. In this task, the robot is required to deliver packages to various offices in the building while maintaining its battery level and periodically checking for newly arrived mail in the drop near the elevators. The task is further complicated by doors that may be locked at different times of the day and office residents who may be available only sporadically. Even a conservative representation of this problem requires well in excess of $2^{500}$ states to fully represent.



- 10 floors
- $>$1800 locations per floor
- $\sim$400 delivery locations
- 25 battery levels
- $\sim$450 doors (some lockable)

Figure 1: Target Domain

**Approaches to Scalable Planning:** While a number of approaches to planning in exponentially large MDPs have been proposed, no single technique has proven powerful enough to completely address problems of the scale described above. Instead, we seek to attack large state spaces by integrating a number of existing techniques with our own novel methods. Currently, we are developing planning systems based on the following methods:

**Hierarchical State Decomposition** State spaces can be decomposed into geographic regions, such as floors, hallways, or rooms, and planning carried out locally rather than globally. The local plans, or macros (represented in Precup's "options" framework [2]), can then be used as "meta-actions" in a higher-level planning operation to produce a global plan.

**Reward Function Decomposition** In many MDP problem domains, the dynamics of the environment (expressed via the state space and transition function) are fixed across problem-solving episodes, but the tasks or goals (expressed by the reward function) are dynamic. We would thus like to amortize planning effort across multiple episodes by developing sub-plans that account for environmental structure but which can be reused in response to current goal values. This leads to an alternate type of decomposition that generates macros for achieving sub-goals of the global reward function. Such macros may also yield a substantial improvement in one-shot planning effort: sub-goals are often explicitly associated with particular variables of the state space (e.g., a bit may be used to indicate that a particular package has been delivered) and discarding variables irrelevant to the sub-goal can produce an exponential improvement in planning effort for that task. By applying this process to each sub-goal, we produce a set of macros for handling parts of the global task. Again, a higher-level planning process is necessary to integrate the macros into a global plan.

**Deterministic Approximations** Often, the macros that result from a hierarchical or reward decomposition can be effectively treated as deterministic even though the primitive actions are stochastic. When the variance of the effects of a macro (transition time, accumulated reward, etc.) are small relative to the mean, it may be justified to treat it as a deterministic meta-action for the purposes of higher-level planning processes.

**Approximate Meta-Planning** Construction of macros simply removes the planning problem one step; we still need a method for integrating the macro-actions into a global plan. In general, the macro actions form a semi-Markov decision process (SMDP) in a reduced state space, for which standard planning techniques are known [2]. Unfortunately, the meta-state space, while smaller than the primitive space, may only be polynomially so. For example, in the package delivery sub-task of the navigation domain outlined above, goal decomposition yields a space that is exponential in the number of delivery locations and the resulting planning problem is equivalent to the traveling salesdroid problem (TSP). This observation, however, leads us to apply well-known heuristic solution methods for the TSP, in conjunction with deterministic approximations of macro effects, to generate cheap but suboptimal solutions. In some cases, even simpler methods (such as single-step lookahead action selection) have proven to be quite effective.

To date, we have demonstrated the effectiveness of these approaches empirically in small and mid-sized ($\sim 2^{55}$ states) domains. We are currently developing simulations of larger problems, including infrastructure for modeling environments consisiting of arbirary discrete variables and complex, structured transition functions. We have formalized the notion of a reward function decomposition and have demonstrated the construction of complete macro sets for solving linear variants of *isolated* reward components. Handling *multiple* reward components simultaneously depends on the structure of the transition function, and we are currently extending our existing results on shortest path and minimum tour problems to other classes, including maintenance tasks and "rewards of opportunity".

**References:**


[1] C. Boutilier, R. Dearden, and M. Goldszmidt. Stochastic dynamic programming with factored representations. *Artificial Intelligence*, 121(1–2):49–107, 2000.

[2] D. Precup. *Temporal Abstraction in Reinforcement Learning*. PhD thesis, University of Massachusetts, Amherst, Department of Computer Science, 2000.

[3] M. L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, New York, 1994.